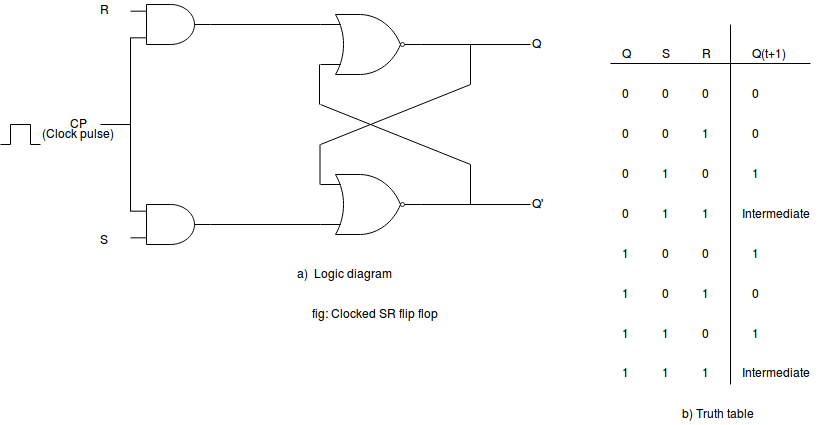
11주차 예비보고서

전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

1. **RS flip-flop에 대해서 조사하시오.**

앞서 우리가 배웠던 기본 논리게이트와 연산을 진행하는 adder및 subtractor는 논리회로이다. 논리회로란 어떤 논리에 의하여 논리 연산을 수행하여 출력값을 얻는 회로를 의미한다. 하지만 이런 논리회로와 다르게 순차회로도 존재하는데 이번 실습에서 구현하고자 하는 flip-flop이 이에 해당한다. 순차회로는 논리회로와 다른 정보를 보관하거나 기억하는 기능을 수행한다. 이러한 기능이 필요한 이유는 단순 연산을 진행하는 거에 더해 이전의 상태를 기억하여 다음 상태에 사용할 수 있도록 하는데 있다. 정보를 저장하여 더 다양한 기능과 출력값을 나타내는 회로를 구성할 수 있다. 이전 정보를 저장하는데 있어 시간도 중요한 역할을 하는데 일정한 clock마다 정보를 갱신하여 출력에 반여하는 것이 flip flop이고 입력에 따라 항상 반영되는 것이 latch이다. 기본 논리게이트처럼 순차회로에도 기본적으로 구성되어 있는 가장 작은 단위의 회로가 있는데 그 중 RS flip-flop에 대해서 알아보자.

RS flip-flop은 Reset-set flip-flop으로 회로의 input으로 Set(S)와 Reset(R)이 들어간다. 또한 flip-flop은 clock 신호에 따라 출력을 반영하므로 clock변수 또한 input으로 들어간다. 여기서 S, R과 clock의 기능을 살펴보자.



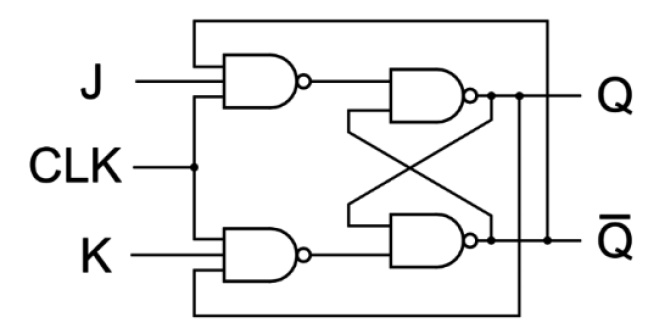
R과 S는 각각 Q의 값을 0 또는 1로 저장하는 역할을 한다. RS=00이면 두 기능 다 작동하지 않아 이전 저장상태를 그대로 유지한다. RS flip-flop에서는 RS=11인 경우 회로가 작동하지 않으며 유효한 정보를 저장하지 못한다. 이를 진리표로 작성하면 다음과 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Qnext | 동작 |
| 0 | 0 | Q | No change |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | X | Invalid |

클럭은 입력을 회로에 반영하여 출력을 나타내는 시점을 정해주는 변수로 flip flop에서는 edge trigger방식으로 사용한다. 클럭은 0 또는 1의 값을 주기적으로 갖는 wave인데 edge trigger방식은 값이 변하는 시점에 신호를 받는 방식을 의미한다.

1. **JK flip-flop에 대해서 조사하시오.**

J와 K는 RS flip-flop에서 RS=11인 경우를 control하기 위해 만든 회로이다. 나머지 경우는 J가 Set, K가 Reset을 하며 작동하고 JK=11인 경우 서로 반대의 값을 갖도록 한다.



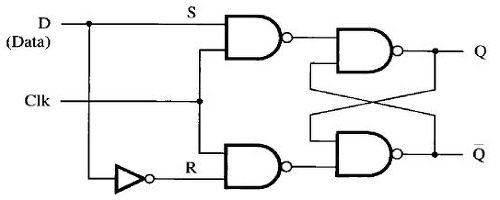
진리표로 작성하면 다음과 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| J | K | Qnext | 동작 |
| 0 | 0 | Q | No change |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | ~Q | Toggle |

여기서 Q와 ~Q는 서로 다른 값을 갖는 보수의 관계의 출력이다. 정보를 저장하여 기억하는 것은 Q이다. JK flip-flop에서는 JK=11인 경우 Q의 값을 ~Q에 저장되어 있던 정보로 저장하는 것을 의미한다. 이때 ~Q는 Q의 값으로 저장된다.

1. **D flip-flop에 대해서 조사하시오.**

D flip-flop은 RS flip-flop을 간소화하기 위해 구성한 회로이다. D flip-flop의 경우 input은 D와 clk두개이다. 회로로 나타내면 다음과 같다.



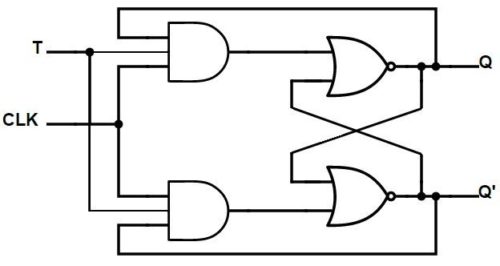
회로를 보면 간단하게 알 수 있다. D의 역할은 S와 R의 역할을 한다. D=0이면 Reset의 동작을 하여 0을 저장하고, D=1이면 Set의 역할을 하여 1을 저장한다. 이를 진리표로 작성하면 다음과 같다.

|  |  |  |
| --- | --- | --- |
| D | Qnext | 동작 |
| 0 | 0 | Reset |
| 1 | 1 | Set |

RS flip flop에서 RS=00, RS=11인 경우를 제외하고 더 간단하게 회로를 구성했다.

1. **T flip-flop에 대해서 조사하시오.**

T flip-flop은 JK flip flop을 간소화하기 위해 구성된 회로이다. JK=00, JK=11의 경우를 통합하여 T로 만들었다.

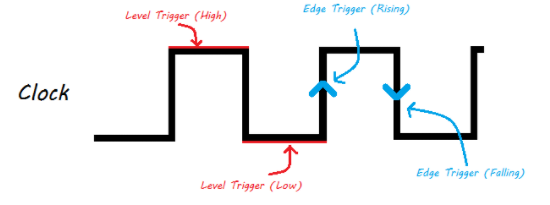


JK의 경우 JK=00일 때, No change의 기능을 하고, JK=11일 때, toggle의 기능을 하는 것을 위에서 알 수 있었다. T는 이 기능을 합하여 T=0일 때, No change기능을 하고, T=1일 때, toggle의 기능을 한다. 이를 진리표로 나타내면 다음과 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| T | Q | Qnext | 동작 |
| 0 | 0 | 0 | No change |
| 1 | 0 | 1 | Toggle |
| 0 | 1 | 1 | No change |
| 1 | 1 | 0 | Toggle |

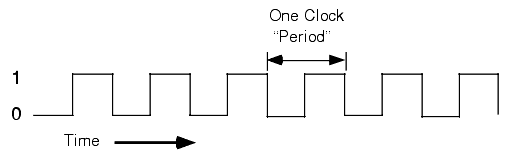
1. **Latch의 기능에 대해서 조사하시오.**

Latch는 flip-flop과 입력을 반영하는 방식에서 차이가 있다. Flip flop의 경우 위에서 언급한 것처럼 clock을 사용하여 edge-trigger방식으로 입력을 반영한다.

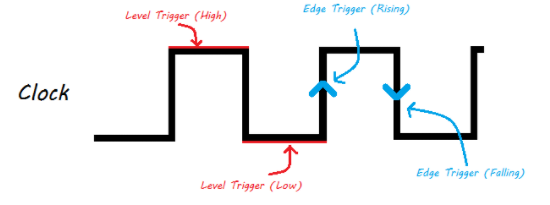


Clock을 통해 입력을 회로에 반영하여 출력을 나타내는 것을 동기 처리라 한다. Latch는 입력 값이 바뀌면 항상 출력을 나타내는 회로로 clock을 사용하지 않고 비동기식 처리를 한다. 이는 FPGA처럼 큰 회로를 갖는 기계에서 동기식, 비동기식 회로를 함께 사용하므로 Latch 또한 필요하다.

1. **Clock의 기능에 대해서 조사하시오.**



Clock은 시간에 따라 주기적으로 High와 Low값을 갖는 신호이다. Clock 신호에 따라 입력을 회로에 반영하여 출력을 나타낼 수 있다. Clock의 신호를 사용하는 방법은 두가지가 있는데, level trigger와 edge trigger가 있다. Level trigger는 1 또는 0의 값을 갖는 동안 입력을 회로에 반영하는 방식이고, edge-trigger의 경우 값이 0에서 1로 바뀌거나 1에서 0으로 바뀌는 시점에 회로에 반영하는 방식이다.

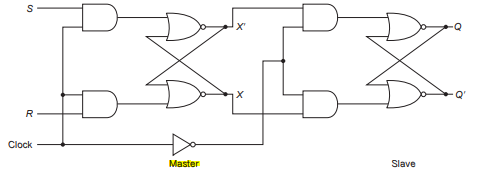


1. **Edge-Trigger의 특성에 대해서 조사하시오.**

Edge-trigger 방식은 위의 설명한 것과 같다. Edge-trigger의 특성에는 level trigger와 다르게 특정 시점에만 신호를 준다는 점이다. Level trigger의 경우 값이 유지되는 동안 신호도 유지한다. 하지만 edge trigger의 경우 rising, falling하는 시점에 신호가 발생하고 유지되지 않는다. T flip-flop에서 T=1일 때, toggle의 기능을 하는데 clock신호에 따라 입력을 반영하기 때문에 만약 신호가 유지되고 있는 상태이면 Q의 값이 신호가 유지되는 동안 계속 toggle되는 상황이 생긴다.

1. **Master-Slave의 개념에 대해서 조사하시오.**

위의 언급한 계속 toggle되는 상황을 방지하기 위해 고안된 회로이다. Master slave 회로는 두 개의 순차회로를 직렬로 연결하여 clock신호가 유지되어도 toggle이 반복되는 상황을 제어할 수 있다. 회로를 먼저 보자.



Clock 신호와 하나의 순차회로가 있고, clock의 inverter를 통해 다른 하나의 순차회로와 연결되어 있다. Clock에 신호가 들어오면 앞의 master 회로가 작동하며 값을 저장한다. 하지만 이때, slave의 회로는 작동하지 않는다. 이후 clock신호가 없을 때, master는 작동하지 않고, slave만 작동하여 결과를 나타낸다. 이는 clock신호가 유지되어도 slave가 작동하지 않아 정보는 master에만 저장되며, 신호가 꺼지는 시점에 slave가 작동하여 출력에 반영한다.

1. **기타이론.**

회로에는 Gate delay가 있어 회로에 입력이 들어가도 바로 출력이 나타나지 않는다. 이는 순차회로에서도 발생한다. 따라서 Clock 신호가 들어와도 그 즉시 출력이 반영되지 않고 delay 이후 입력에 따른 출력이 반영된다. 회로가 커지면 다양한 회로가 조합되고, clock신호도 여러 개 사용될 수 있는데 이때, delay를 무시할 수 없는 상황이 생기게 된다. clock신호가 발생했을 때, input값이 변화하거나 clock 신호에 따라 값이 변화한 후 유지하지 못하면 정상적인 출력을 할 수 없다.